Rajagopal, K., Zhang, Q., Balakrishnan, S. N., Fakhari, P., & Busemeyer, J. R. (to appear). Quantum amplitude amplification for reinforcement learning. In K. G. Vamvoudakis (Ed) *Handbook on Reinforcement Learning and Control.* Springer Studies in Systems, Decision and Control

Quantum amplitude amplification for reinforcement learning

K. Rajagopal, Q. Zhang, S. N. Balakrishnan, P. Fakhari, and J. R. Busemeyer

[1] K. Rajagopal, and Q. Zhang and S. N. Balakrishnan Missouri University of Science and Technology, Rolla, Missouri bala@mst.edu
[2] J. R. Busemeyer and P. Fakhari Indiana University, Bloomington, Indiana, jbusemey@indiana.edu

Summary. *Reinforcement learning models require a choice rule for assigning probabilities to actions during learning. This chapter reviews work on a new choice rule based on an amplitude amplification algorithm originally developed in quantum computing. The basic theoretical ideas for amplitude amplification are reviewed, as well as four different simulation applications to a task with a predator learning to catch a prey in a grid world, and one application to human learning in a 4 arm bandit choice task. The applications reviewed in this chapter demonstrate that QRL can improve speed and robustness of learning compared to standard choice rules such as epsilon-greedy and softmax choice rules.

Summary. Reinforcement learning models require a choice rule for assigning probabilities to actions during learning. This chapter reviews work on a new choice rule based on an amplitude amplification algorithm originally developed in quantum computing. The basic theoretical ideas for amplitude amplification are reviewed, as well as four different simulation applications to a task with a predator learning to catch a prey in a grid world, and one application to human learning in a 4 arm bandit choice task. The applications reviewed in this chapter demonstrate that QRL can improve speed and robustness of learning compared to standard choice rules such as epsilon-greedy and softmax choice rules.

1 Exploration and exploitation in reinforcement learning

Researchers across the world are racing to build quantum computers because of their great potential to speed up computational processing. For example, the Shor algorithm [18] can give an exponential speedup for factoring large integers into prime numbers, and the Grover algorithm [7] can achieve quadratic speedup in search on a quantum computer as compared to classical algorithms on a classical computer. Quantum walks perform a diffuse search at a rate of time squared as opposed to the classical rate of diffusion that is proportional to time [11]. These improvements in computational speed depend on the construction of quantum computers of sufficient size to make a practical advantage over classical computers. However, some researchers are beginning to examine the use of quantum algorithms on classical computers to improve computational speed and performance over classical algorithms [17]. The purpose

of this chapter is to explore the application of an algorithm related to the Grover algorithm to reinforcement learning.

Generally speaking, reinforcement learning (RL) algorithms have two separable parts: one is the learning algorithm that updates the future rewards expected to be produced by taking an action from a current state that leads to a new state (e.g., the Q value for a state and action); and the second is the probabilistic choice rule for selecting an action given a state [21]. The latter moderates what is often called the exploration / exploitation properties of a RL algorithm. It is critical for efficient learning of optimal solutions to have the appropriate balance of exploration and exploitation. This chapter is primarily concerned with the type of probabilistic choice rule used in reinforcement learning. Two of the most commonly used types are the epsilon-greedy choice rule and the logistic (soft max) choice rule [21].

This chapter reviews recent work examining a new kind of choice rule based on a well-known quantum probability search algorithm called amplitude amplification [10], which is a generalization of the Grover algorithm. This algorithm was originally adapted for reinforcement learning [4, 3] and we [2, 5] developed an improved version of the quantum algorithm for reinforcement learning. Simulation studies have demonstrated faster learning using the amplitude amplification algorithm as compared to the epsilon-greedy [4, 3] and the soft max [5] choice rules. Very recently, empirical evidence has found that a learning model based on amplitude amplification predicts human learning on a 4 arm bandit task better than traditional models using a soft max type of rule [14]. The purpose of this chapter is to (a) briefly introduce quantum probability theory, (b) review the quantum reinforcement learning algorithm, and (c) review simulation work that has been used to establish its faster and more efficient learning properties.

2 Quantum probability theory

Before presenting the amplitude amplification algorithm, it is useful to first describe some of the basic principles of quantum probability theory [8]. Although quantum probability was originally designed for physics, recently the mathematical principles have been applied outside of physics to human judgment and decision making [2, 12], information retrieval [22, 16], artificial intelligence and machine learning [15], social and economic science [9, 24] and finance [1].

Classical probability theory evolved over several centuries, however, an axiomatic foundation was first developed by Kolmogorov [13]. Kolmogorov theory is founded on the premise that events are represented as subsets of a larger set called the sample space. Quantum mechanics was invented by a brilliant group of physicists in the 1920's, which revolutionized our world. However, not until von Neumann [23] provided an axiomatic foundation did physicists realize that they actually invented an entirely new theory of probability. Quantum probability is founded on the premise that events are represented as subspaces of a vector space (called a Hilbert space, which is a complete inner produce vector space defined on a complex field). Each subspace of a vector space corresponds to a projector that projects vectors onto the subspace. The basic axioms of quantum probability can be listed as follows.

1. An event A is represented by a subspace in the Hilbert space, which corresponds to a projector $\mathcal{P}_A$ for event A, where $\mathcal{P}_A = \mathcal{P}_A^{\dagger} = \mathcal{P}_A \cdot \mathcal{P}_A$. For example, an event is the choice of an action, and a projector will later be used to represent the choice of an action.
2. The state of a system is represented as unit length vector, symbolized as $|\psi\rangle$, in the Hilbert space. For example, this state vector will later represent the potentials for an agent to take each of several actions.

3. The probability of an event A equals the squared projection: $p(A) = \|\mathcal{P}_A \cdot |\psi\rangle\|^2$. For example, this is how we will later compute the probability of choosing an acton.
4. If two events, A, B, are mutually exclusive, then $\mathcal{P}_A.\mathcal{P}_B = 0$, and the projector for the event that either A or B occurs is $\mathcal{P}_A + \mathcal{P}_B$, and probability of either event is the sum : $p(A \vee B) = \|(\mathcal{P}_A + \mathcal{P}_B) \cdot \psi\|^2 = \|\mathcal{P}_A \cdot \psi\|^2 + \|\mathcal{P}_B \cdot \psi\|^2$.
5. If event A is known to occur, then the conditional probability of event B equals $p(B|A) = \frac{\|P_B \cdot P_A \cdot \psi\|^2}{\|P_A \cdot \psi\|^2}$.

As noted above in item 4, the quantum algorithm is an additive measure: that is, it is a probability measure p such that if A, B are mutually exclusive events, then $p(A \text{ or } B) = p(A) + p(B)$. More important, any algorithm for assigning probabilities to subspaces greater than 2 dimensions in a vector space that satisfy an additive measure will turn out to be equivalent to the quantum algorithm [6].

The beauty of using a vector space is that there are an infinite number of bases that can be used to represent the state and the subspaces. Once a basis is selected, the state can be expressed in terms of that basis. Suppose the Hilbert space has dimension N, with a basis $\{|V_j\rangle, j = 1, \cdots, N\}$. Then we can expand $|\psi\rangle$ in this basis as follows $|\psi\rangle = \sum \psi_j \cdot |V_j\rangle$. In other words, we can represent the state vector $|\psi\rangle$ as the $N \times 1$ matrix ψ with coordinate ψ_j corresponding to basis vector $|V_j\rangle$. We can also represent the projector $\mathcal{P}_A$ in this same basis by an $N \times N$ matrix P_A. For example, the event A could be a one dimensional subspace (a ray) spanned by V_3 and the projector for this event is the outer product $V_3 \cdot V_3^\dagger$

We might, however, find it necessary to represent the event B in a different basis $\{|W_j\rangle = \mathcal{U} \cdot |W_j\rangle, j = 1, \cdots, N\}$, where $\mathcal{U}$ is a unitary operator $\mathcal{U}^\dagger \cdot \mathcal{U} = \mathcal{I}$. In this case, the events may not commute, $P_A \cdot P_B - P_B \cdot P_A \neq 0$, in which case we say they are incompatible. For example, the event B could be a one dimensional subspace (a ray) spanned by W_3 and the projector for this event is the outer product $W_3 \cdot W_3^\dagger$. Note that $V_3 \cdot V_3^\dagger$ does not commute with $W_3 \cdot W_3^\dagger$. If the events A, B are incompatible, then $P_A \cdot P_B$ is not a projector, and $p(A) \cdot p(B|A) \neq p(B) \cdot p(A|B)$, and there is no joint probability for events A, B.

We can represent the unitary operator, $\mathcal{U}$, in the V basis by another $N \times N$ matrix U. Note that the unitary transformation preserves lengths and inner products. Later, the Grover algorithm is expressed by a unitary transformation.

So far we have only described the structural part of quantum probability theory. The dynamic part of quantum theory forms another important part. For example, quantum walks use the dynamic part of the theory. Essentially, the dynamics describe how the unitary transformation changes with time. However, for the application considered here, we only need the structural part of the theory, and so we will not present the dynamic part.

3 The original quantum reinforcement learning (QRL) algorithm

Consider the popular Q-learning algorithm. Define the estimate of the expected discounted future rewards for each state and action at time t as $Q(s_i, a_k, t)$. Suppose the last action taken at time t changed the state from s_i to state $s_{i'}$ and the immediate reward $r(t)$ was obtained. Then the new estimate for each state is updated according to

$$Q(s_i, a_k, t+1) = (1-\eta) \cdot Q(s_i, a_k, t) + \eta \cdot \left[r(t) + \gamma \cdot \max_l Q(s_{i'}, a_l, t) \right], \tag{1}$$

where η is a learning rate parameter and γ is a discount rate parameter. The term in square brackets is the Q-learning 'reward' signal. Then the next action is probabilistically selected based on its expected future reward value. Two commonly used methods to probabilistically choose an action are the epsilon-greedy rule and the softmax rule. The epsilon-greedy rule selects the best action with probability $(1-\epsilon), 0<\epsilon<1$, and otherwise randomly selects an action. The softmax rule selects action k with probability $p(A_k) = e^{Q_k/\tau}/\sum e^{Q_j/\tau}$, where τ is called the temperature.

Quantum reinforcement works as follows. At any given state of the environment, suppose the agent has m possible actions. Then the dimension of the Hilbert space is set equal to m, and the basis is chosen such that each orthonormal basis vector represents an action. The state can then be represented by a $m \times 1$ unit length state vector denoted ψ, with a coordinate (called an amplitude) ψ_k corresponding to the basis vector for action a_k.The projector for an action can be represented by a $m \times m$ diagonal indicator matrix that simply picks out the coordinate corresponding to the action. Finally, the probability of taking action a_k simply equals $|\psi_k|^2$. The key new idea is the rule for amplifying the amplitudes ψ_k based on the expected future value of an action a_k from the current state s_i after some number of training trials t denoted $Q(s_i, a_k, t)$.

The amplitude amplification algorithm is an extension by Brassard and Hoyer [10] of Grover's [7] search algorithm. The algorithm begins with an initial amplitude distribution represented by the $m \times 1$ matrix ψ_0, with $\psi_k = \frac{1}{\sqrt{m}}$. Define ψ_1 as the $m \times 1$ matrix of amplitudes after experiencing amplification on a trial.

Suppose action a_j was chosen on the last trial. In the original QRL algorithm [4, 3], the amplitude for action a_j is amplified or attenuated in proportion to the reward signal $[r(t) + \gamma \cdot \max_l Q(s_{i'}, a_l, t)]$ experienced by taking that action. This is done as follows.

Define A_k as an $m \times 1$ matrix with zeros in every row except the row k corresponding to action a_k, which is set equal to one. Next define the following two unitary matrices

$$U_1 = I - \left(1 - e^{i\phi_1}\right) \cdot \left(A_k \cdot A_k^{\dagger}\right), \tag{2}$$
$$U_2 = \left(1 - e^{i\phi_2}\right) \cdot \left(\psi_0 \cdot \psi_0^{\dagger}\right) - I$$

where ϕ_1, ϕ_2 are two learning parameters that control the amount of amplification or attenuation. The matrix Q_1 flips the sign of the target action, and the matrix Q_2 inverts all the amplitudes around the average amplitude, and together these to act to amplify the target while having no effect (except normalization) on the non targets. Then the new amplitude distribution is formed by

$$\psi_1 = (U_2 \cdot U_1)^L \cdot \psi_0, \tag{3}$$

where the matrix power L indicates the integer number of applications of the update used on a single trial.

The original QRL algorithm fixed $\phi_1 = \phi_2 = \pi \approx 3.1416$, and related the reward $[r(t) + \gamma \cdot \max_l Q(s_{i'}, a_l, t)]$ to L applied after each trial. The parameter L was set equal to the integer value of $c \cdot [r(t) + \gamma \cdot \max_l Q(e, a_l, t)]$ where $c > 0$ is a free parameter [4]. This essentially produces the original Grover updating algorithm.

4 The revised quantum reinforcement learning algorithm

The original QRL, based on relating the parameter L to rewards, restricts the algorithm to discrete jumps in amplitudes, and it is too restrictive for small numbers of actions, and it does

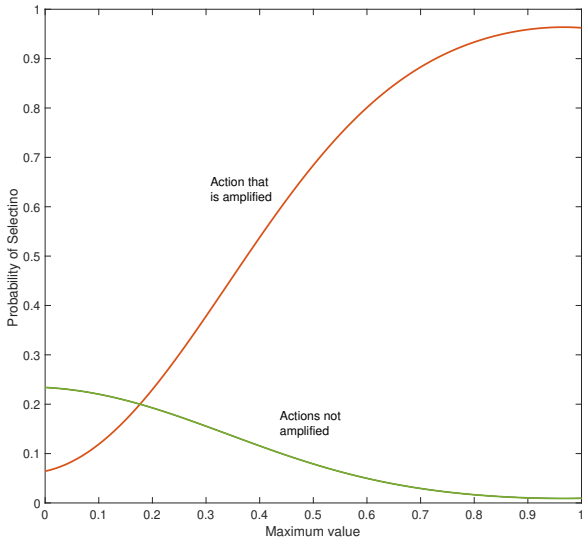
not provide a good method for treating punishments. Another option first proposed by [2] and then improved in [5] was to set $L = 1$ and vary ϕ_1 and ϕ_2 as described below

1. Find the action a_{max} corresponding to the maximum Q-value in the current state i and select it for amplitude amplification.
2. Find the set of next possible states, denoted S, from the current state i, and the next state i' produced by a_{max}, and then compute the ratio

$$\phi = \frac{\max_{l} Q(s_{i'}, a_l, t)}{\max_{s'} \max_{l} Q(s_{i'}, a_l, t)} \tag{4}$$

   If $\max_{s'} \max_{l} Q(s_{i'}, a_l, t) = 0$ or $\phi < 0$ then $\phi = 0$.
3. Set $\phi_1 = \pi \cdot (a \cdot \phi + b)$ and $\phi_2 = c \cdot \phi_1$. We found $a = 1.3$,$b = 15.8$, and $c = .65$ to work well based on the relation between choice probability and normalized Q value shown in Figure 1.
4. The apply Equations 1,2 using ϕ_1, ϕ_2, and $L = 1$.

Fig. 1. Relation between choice probability for each action and normalized Q value using parameters described in text



5 Learning rate and performance comparisons

The first investigation of quantum reinforcement learning (QRL) [4] conducted simulations of a predator-prey task using a 20 by 20 square grid world. Each cell represented a state with a

start state in the upper left corner and a goal state in the lower right corner, and some cells designed as blocks that prevented movement. From any cell, the agent could move up, down, left, right. The reward for finding the goal was 100 points, and each step cost 1 point. This investigation used a TD(0) algorithm with an epsilon greedy choice rule $\epsilon = .01$ and compared the QRL chioce rule, using the orignal amplitude amplification rule. They varied the learning rate parameters of the TD(0) algorithm. Their simulation results revealed that generally the QRL rule explored more initially but learned to reach the goal faster than the epsilon-greedy algorithm based on the number of episodes required to reach the goal.

The second investigation of QRL [3] used collection of 58 states with an arrangement of permissible transitions to new states from a given state. Once again the agent started in some starting state and transitioned at a cost of 1 point each step to a goal state worth 100 points. In this investigation, the agent had to learn to find the goal with different starting positions. (Although it is not explicitly stated, we assume that an epsilon-greedy algorithm was again used for the traditional RL model, and the QRL was based on setting $\phi_1 = \phi_2 = \pi$ and selecting L based on the reward signal). The simulation results showed that the QRL algorithm learned faster and was more robust than the traditional RL model when changing starting positions, and the results were robust across a range of learning rate parameters. The QRL and traditional RL models were also compared using a simulated robot navigation task where the goal was to avoid obstacles. The robot was endowed with 16 sensors to provide sensory data on the obstacles in the environment. Each step cost one point, reaching the goal produced 100 points, but hitting an obstiacle stopped the episode and produced a loss of 100 points. Once again, the agent using the QRL algorithm learned faster to avoid obstacles than the traditional RL agent.

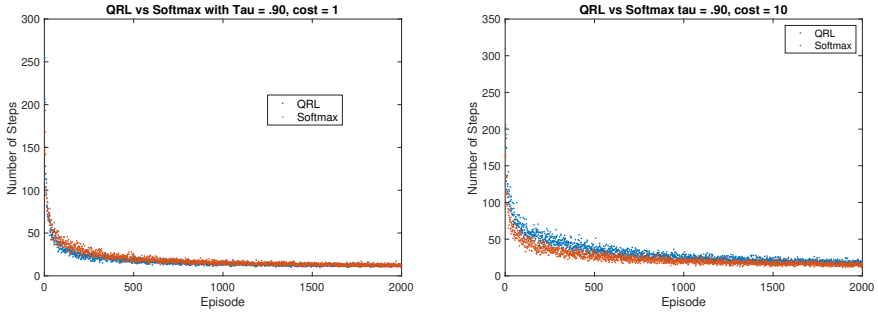
The third investigation of QRL [5] used simulations to investigate three different types of predator - prey tasks. The first task used a 20 by 20 square grid world with obstacles in which the goal was fixed throughout an episode (defined from start until goal capture or stop search in 4000 steps). The state was defined by a cell, and the predator had five actions (left, right, up, down, stay in place). The agent started in different states, and the reward for catching the goal was 100 with a cost of one point per step were examined. Both an epsilon-greedy ($\epsilon = .01$) and a softmax rule ($\tau = .9$) were examined. The QRL model was based on the revised amplitude amplification algorithm. They also simulated the same paradigm with softmax and quantum algorithms for different grid world sizes for 5 actions and 9 actions. The QRL model again produced faster learning than both the epsilon - greedy and softmax standard RL models.

The second task used a 10 by 10 grid world without obstacles. However, in this case both the predator and the prey started in arbitrary states and moved from step to step with a 100 point reward for catching the goal and a cost of one point per step. The prey moved randomly on each step. For this task, a state was defined by the vertical and horizontal steps needed to reach the prey. Again, the QRL model again produced faster learning than both the epsilon - greedy and softmax standard RL models.

Figure 2 (left panel) illustrates the learning results for the QRL and the softmax rule ($\tau = .9$). A total 100 simulations were run, and each simulation contained 5000 episodes. The predator and the prey started in arbitrary states and both moved from step to step. For this task, a state was defined by the vertical and horizontal steps needed to reach the prey. The panel in the left of the figure shows the average over 100 simulations of learning rates for the QRL and softmax ($\tau = .9$) rule across 2000 episodes. Similar results are obtained with $\tau = .5$ and 1.5.

The faster learning by the QRL compared to the softmax rule is not, however, universal. As shown in Figure 2 right panel, if we increase the cost of each step from 1 to 10, then the softmax rule learns faster than the QRL rule. Nevertheless, as we next , the QRL still performs better in a competition with softmax after 20000 episodes of training.

Fig. 2. Number of episodes required to catch prey for QRL and Softmax (τ = .90)



Finally, a third task was investigated that directly pit a QRL agent against a softmax agent for chasing and reaching the prey. In this case, a 10 by 10 grid world was used, and both the predator and prey moved on each step (the prey moved randomly). For this task, a state was again defined by the vertical and horizontal steps needed to reach the prey. Catching the prey produced a reward equal to 100 points, and each step cost one point. Both agents were initially trained with just the prey for 20,000 episodes using either the softmax rule (τ = .9) or the revised amplitude amplification rule. After this training, the learning algorithm was turned off and the choice at each state was based on the maximum Q value. Both agents started at the same position at the upper left corner, and both competed within the same episode to catch the prey. Each episode ended with either the QRL agent catching the prey first, the softmax agent catching the prey first, or both agents catching the prey at the same time. The results were that the QRL agent substantially beat the softmax agent to catch the moving prey.

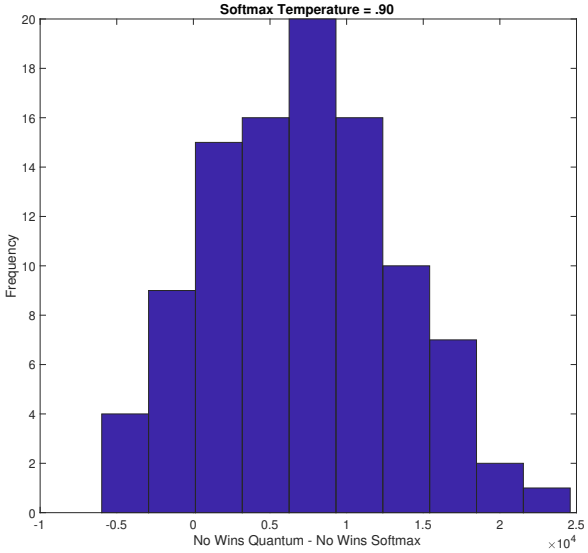
We re-ran these simulations with different values for the temperature (τ = .15, .2, .3, .4, .5, .6, .7, .8, .9, 1.5, 2.0), which all produced similar results. Figure 3 shows the results produced by simulating 100 pairs of QRL and softmax agents (τ = .9). Each pair was tested with 20,000 preys (episodes with a different randomly generated prey for each episode.) The performance measure was the number of times out of 20,000 that the QRL agent won minus the number of times the softmax agent won. As can be seen in the figure, the QRL agent most frequently beat the softmax agent by a substantial amount.

We also re-ran these simulations with different costs for each step (1, 5, and 10). Table shows the quartiles of the distributions for a cost of 10 points per step. As can be seen, the QRL distribution dominates the softmax distribution for number of wins. Similar results were obtained with costs equal to 1 and 5. It is interesting that even though the softmax learned faster with a cost of 10 points, the QRL performed better after training.

Table 1. Quartiles from 100 samples of the number of times QRL won, softmax (τ = .90) won, and ties, out of 20000 episodes for a cost of 10 points per step by predator.

	QRL wins	Softmax wins	choice rule tie
25%	1.7295	1.6252	0.9030
50%	1.9547	1.7742	1.1171
75%	2.2495	1.9612	1.5700

Fig. 3. Frequency out of 100 pairs for each category representing the number of times QRL agent won minus number of times softmax agent won



6 Other Applications of QRL

The amplitude amplification algorithm was also applied to fault detection and parameter identification for stochastic dynamic systems [25]. Detection of faults and adaptive estimation of parameters are crucial to successful control of vehicles. When a fault occurs, the parameters of a system change and can change drastically; in order to cope with such changes, a quantum-boost scheme to work with a multiple-model Kalman filter was developed [25]. The basic estimation method is based on a multiple-model Kalman filter. At each time step, the a priori estimation and covariance of the Kalman filters are weighted with the conditional probability of the parameter set, given the new measurement. The conditional probability is obtained by Bayesian inference. The conditional probability of the parameter set closest to the correct parameter set typically rises with new information and finally converges to one. In fault detection, it is of vital significance that such probability rise and convergence happen quickly. This can be achieved with a quantum amplitude amplification boost scheme. The conditional probability is updated twice at each time step, once with Bayesian inference and once with extended amplitude amplification algorithm. The probabilities of different parameter sets can be considered as weights for the parameter sets. The weighted estimation can be considered as a superposition state and the estimations of the sub-filters with different parameter sets as basis states. Then the probabilities are the square of the amplitude of the basis states. The amplitude amplification algorithm is suitable for probability update because the operator is unitary and hence the updated probabilities will sum up to 1. A proof to assure that the boost under certain conditions can accelerate the convergence of probabilities can be found in [26].

Consider the following linear system:

$$x_k = F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1} \tag{5}$$

$$y_k = H_k x_k + v_k \tag{6}$$

$$w_k \sim N(0, Q_k) \tag{7}$$

$$v_k \sim N(0, R_k) \tag{8}$$

where the subscript k indicates the k'th time step, x_k is an n-dimensional state vector, y_k is a q-dimensional measurement vector, u_k is an m-dimensional control input, w_k is an n-dimensional system noise vector with a covariance given by Q_k; v_k is q-dimensional measurement noise vector with a covariance of R_k. All noise sequences are assumed normal. G and H are appropriately dimensioned matrices. Parameter vector p is defined as the set (F, G).

For derivations of a typical multiple-model Kalman filter, the reader is referred to [19, 20].

The quantum-boost multiple-model Kalman filter algorithm is given as follows:

Input: Initialize probabilities for parameter p_j given measurement $\Pr(p_j|y_0)$ (j = 1, ..., N), the initial a posterioi state estimate, $\hat{x}_{0j}^-$ and the corresponding a posteriori covariance, P_{0j}^-.

At each time step k>0, perform the following steps:

1. Baysian inference for multiple-model Kalman filter

a) Run N Kalman filters for each p_j. The a priori state estimate and covariance (indicated with minus sign in superscript) of jth filter at step k are computed from a posteriori state estimate and covariance (indicated with plus sign in superscript) at step $k-1$:

$$\hat{x}_{kj}^- = F_{k-1,j}\hat{x}_{k-1}^+ \tag{9}$$

$$P_{kj}^- = F_{k-1,j}P_{k-1}^+F_{k-1,j}^T + Q_{k-1} \tag{10}$$

b) After the kth measurement, approximate $\mathrm{pdf}(y_k|p_j)$ is given by

$$\mathrm{pdf}(y_k|p_j) \approx \frac{\exp(-r_k^T S_k^{-1} r_k/2)}{(2\pi)^{q/2}|S_k|^{1/2}} \tag{11}$$

where $r_k = y_k - H_k\hat{x}_{kj}^-$ is the residual, $S_k = H_k P_{kj}^- H_k^T + R_k$ is the covariance, q is number of measure and $\mathrm{pdf}(\cdot)$ refers to the probability density function of $(\cdot)$.

c) Estimate the probability that $p = p_j$ as follows.

$$\Pr(p_j|y_k) = \frac{\mathrm{pdf}(y_k|p_j)\Pr(p_j|y_{k-1})}{\sum_{i=1}^N \mathrm{pdf}(y_k|p_i)\Pr(p_i|y_{k-1})} \tag{12}$$

where $\Pr(\cdot)$ represents the probability of $(\cdot)$.

2. Quantum boost for multiple-model Kalman filter

a) Prepare the following superposition state

$$|\psi\rangle = \sum_{j=1}^{N} \sqrt{\Pr(p_j|y_k)}\,|j\rangle \tag{13}$$

where the canonical basis $|j\rangle$ is simply an $N \times 1$ vector where all elements are zero except the jth element is one, and $|\psi\rangle$ is a superposition state.

b) Find p_j with greatest probability rise

$$J = \mathrm{argmax}_j(\Pr(p_j|y_k) - \Pr(p_j|y_{k-1})) \tag{14}$$

c) Select $|J\rangle$ as the qubit to be boosted.

d) Calculate the unitary extended Grover operator $G = U_2 U_1$ according to Eq. 2.
e) $|\psi\rangle \leftarrow G|\psi\rangle$
f) $\Pr(p_j|y_k) \leftarrow |A_J|^2$ where A_J is the complex coefficient of $|J\rangle$.
3. Combine the subfilters into the multiple-model Kalman filter
a) Weight each $\hat{x}_{kj}^-$ and P_{kj}^- accordingly to obtain

$$\hat{x}_k^- = \sum_{j=1}^{N} \Pr(p_j|y_k)\hat{x}_{kj}^- \tag{15}$$

$$P_k^- = \sum_{j=1}^{N} \Pr(p_j|y_k)P_{kj}^- \tag{16}$$

b) Estimate the a posteriori state estimate and covariance as

$$\hat{x}_k^+ = \hat{x}_k^- + K_k(y_k - H_k\hat{x}_k^-) \tag{17}$$

$$P_k^+ = (I - K_k H_k)P_k^- \tag{18}$$

The Grover update is implemented only once at each time step. We use $\phi_1 = \pi\phi$ and $\phi_2 = 0.15\phi_1$ so that the probability increases monotonically as ϕ increases [5]. ϕ is carefully chosen so that the probability is increased while not causing divergence.

6.1 Example

Efficacy of the quantum-boosted multiple-model Kalman filter was tested with two examples. In both examples, the quantum boost scheme was seen to accelerate the rise and convergence of the unknown initial parameter and also the changed parameter. The system equation and measurement equation of the first example (harmonic oscillator) are as follows [19, 20].

$$\dot{x} = Ax + Bw_1 = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} x + \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix} w_1 \tag{19}$$

$$y_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x_k + v_k \tag{20}$$

$$w_1 \sim N(0, Q_c) \tag{21}$$

$$v_k \sim N(0, R) \tag{22}$$

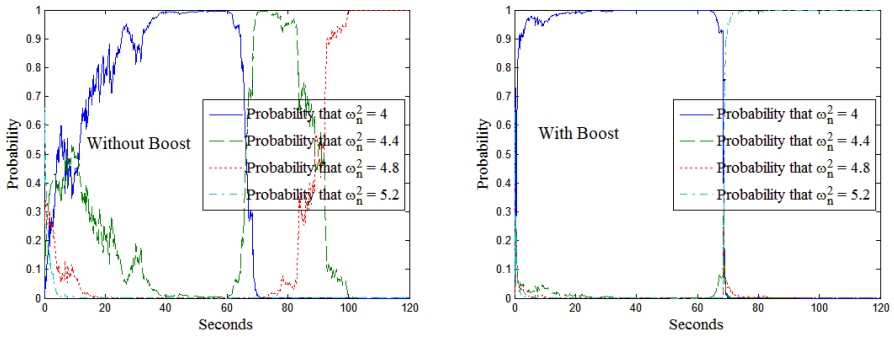
The damping ratio $\zeta = 0.1$. Process and measurement noise covariance are set at $Q_c = 1000$ and $R = 10I$, where I is a 2×2 identity matrix.

Value of the natural frequency ω_n is assumed unknown to the multiple-model filter. The objective is to find the true natural frequency ω_n of the system in Eq. 19. In this example, four Kalman filters are carried with different values of ω_n^2: 4.0, 4.4, 4.8, and 5.2 with a priori probabilities $\Pr(\omega_n^2 = 4.0) = 0.1$, $\Pr(\omega_n^2 = 4.4) = 0.2$, $\Pr(\omega_n^2 = 4.8) = 0.3$, and $\Pr(\omega_n^2 = 5.2) = 0.4$.

In the system equation, ω_n retains its true normal value of 2.0 ($\omega_n^2 = 4.0$) when $t = 0 \sim 60$ sec and a fault is assumed to happen changing ω_n to an abnormal value ($\omega_n^2 = 5.6$) at $t = 60$ sec.

Results are presented where the simulations were carried out with a multiple-model filter without a quantum boost and with a quantum boost and their relative performance in parameter estimation and fault detection area compared.

Fig. 4. Histories of the probabilities of different ω_n^2



Histories of the probabilities of different ω_n^2 in the first example of harmonic oscillator (ω_n is natural frequency) are shown in Fig. 4. Note that without the quantum boost, $\Pr(\omega_n^2 = 4)$ (the probability of the correct ω_n^2) converges to 1 in 40 sec. With the quantum boost, $\Pr(\omega_n^2 = 4)$ converges to 1 in 20 sec. In the case of fault detection, there is a fault at t =60 sec and ω_n^2 jumps to 5.6; since the right value is not amongst the ones assumed, the best that can happen is to reach the value closest to the right value, which in this $\omega_n^2 = 5.2$. In the simulations without the quantum boost, $\Pr(\omega_n^2 = 5.2)$ does not show any increase in the next 60 sec of the simulation; With the quantum boost, $\Pr(\omega_n^2 = 5.2)$, however, converges to 1 in just 15 sec after the fault has occurred. Note that by switching the parameter values around the newly converged value and carrying the same number of filters, we can converge to the true value. This is shown in an upcoming paper. With quantum-boosted multiple-model Kalman filter, fast fault detection can be achieved by monitoring the probabilities of the assumed values of the parameters.

7 Application to human learning

Recently, a QRL model was applied to human decision making with 100 participants trained for 180 trials on a 4 arm bandit type of reinforcement learning task [14]. Two different QRL model were compared with twelve traditional RL models based on the trial by trial behavioral choices. In addition, the relation between estimated model paramaters and functional magnetic resonance imaging (fMRI) was examined. The QRL models performed well compared with the best traditional RL models. The internal state representation from the QRL model was also related to fMRI brain activity measures in the medial frontal gyrus. This was the first study to relate quantum decision processes to neural substrates of human decision making.

8 Concluding comments

The trade-off between exploration and exploitation is critical for reinforcement learning theory, which is an issue concerning the assignment of probabilities to actions during learning. This chapter reviews work on a new method based on an amplitude amplification algorithm originally developed in quantum computing. The new method is called quantum reinforcement learning,

however, it still uses traditional learning algorithms, such as Q-learning, and only modifies the choice rule for selecting actions based on the learned expectations. The applications reviewed in this chapter demonstrate that QRL can improve speed and robustness of learning compared to standard choice rules such as epsilon-greedy and softmax choice rules. However, more work is needed to investigate the range of learning situations where this advantage is found, and also more work is needed to compare other types of choice rules (such as for example, simulated annealing). Finally, amplitude amplification employs only a very small part of quantum computing tools, and other quantum applications, such as for example, quantum walks for modeling flow of information in networks [17] , may provide additional engineering contributions.

References

1. B. E. Baaquie. Quantum mechanics and option pricing. In *Proceedings of the Second Quantum Interaction Symposium (QI-2008)*, pages 54–59, 2008.
2. J. R. Busemeyer and P. D. Bruza. *Quantum models of cognition and decision*. Cambirdge University Press, 2012.
3. D. Dong, C. Chen, J. Chu, and T. J. Tarn. Robust quantum inspired reinforcement learning for robot navigation. *IEEE/ASME Transactions on Mechatronics*, xx:1–12, 2010.
4. D. Dong, C. Chen, H. Li, and T. J. Tarn. Quantum reinforcement learning. *IEEE Transactions Systems Man Cybernetics, B: Cybernetics*, 38 (5):1207–1220, 2008.
5. Rajagopal K. Balakrishnan S. N. & Busemeyer J. R. Fakhari, P. Quantum inspired reinforcement learning in changing environments. *New Mathematics and Natural Computation: Special Issue on Engineering of the Mind, Cognitive Science and Robotics*, 9(3):273–294, 2013.
6. A. M. Gleason. Measures on the closed subspaces of a hilbert space. *Journal of Mathematical Mechanics*, 6:885–893, 1957.
7. L. K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Physical Review Letters*, 79 (2):325–327, 1997.
8. S. P. Gudder. *Quantum Probability*. Academic Press, 1988.
9. E. Haven and A. Khrennikov. *Quantum social science*. Cambirdge University Press, 2013.
10. P. Hoyer. Arbitrary phases in quantum amplitude amplification. *Physical Review A*, 62:052304–1 – 052304–5, 2000.
11. Julia Kempe. Quantum random walks: an introductory overview. *Contemporary Physics*, 44(4):307–327, 2003.
12. A. Y. Khrennikov. *Ubiquitous quantum structure: From Psychology to Finance*. Springer, 2010.
13. A. N. Kolmogorov. *Foundations of the theory of probability*. N.Y.: Chelsea Publishing Co., 1933/1950.
14. J. Li, D. Dong, Z. Wei, Y. Liu, P. Yu, F. Nori, and X. Zhang. Quantum reinforcement learning during human decision making. *Nature: Human Behavior*, in press.
15. Wichert A. M. *Principles of quantum artificial intelligence*. World scientific, 2013.
16. Massimo Melucci. *Introduction to information retrieval and quantum mechanics*. Springer, 2015.
17. Eduardo Sánchez-Burillo, Jordi Duch, Jesús Gómez-Gardenes, and David Zueco. Quantum navigation and ranking in complex networks. *Scientific reports*, 2:605, 2012.
18. Peter W Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. Ieee, 1994.

19. Dan Simon. *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.
20. Robert F Stengel. *Optimal control and estimation*. Courier Corporation, 1994.
21. R. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT Press, 1998.
22. K Van Rijsbergen. *The geometry of infomation retrieval*. Cambridge University Press, 2004.
23. J. Von Neumann. *Mathematical Foundations of Quantum Theory*. Princeton University Press, 1932/1955.
24. Alexander Wendt. *Quantum mind and social science*. Cambridge University Press, 2015.
25. Qizi Zhang, Sivasubramanya N Balakrishnan, and Jerome Busemeyer. Fault detection and adaptive parameter estimation with quantum inspired techniques and multiple-model filters. In *2018 AIAA Guidance, Navigation, and Control Conference*, page 1124, 2018.
26. Qizi Zhang, SN Balakrishnan, and Jerome Busemeyer. Parameter estimation with quantum inspired techniques and adaptive multiple-model filters. In *2018 Annual American Control Conference (ACC)*, pages 925–930. IEEE, 2018.